

Supplemental Material to:

**Elie Maza, Pierre Frasse, Pavel Senin, Mondher Bouzayen,
Mohamed Zouine**

**Comparison of Normalization Methods for Differential
Gene Expression Analysis in RNA-Seq Experiments: a
Matter of Relative Size of Studied Transcriptomes**

Communicative & Integrative Biology 2013; 6(6)
<http://dx.doi.org/10.4161/cib.25849>

<http://www.landesbioscience.com/journals/cib/article/25849>

```

#-----
#
# Comparison of Normalization Methods for Differential Gene Expression Analysis
# in RNA-Seq Experiments: a Matter of Relative Size of Studied Transcriptomes.
#
#           - Supplementary Materials -
#
#           E. MAZA, P. FRASSE, P. SENIN, M. BOUZAYEN, M. ZOUINE
#
#           Supplementary File 1 (out of 3).
#
#-----

#-----
# Simulation parameters
#-----

G <- 30000
M.Up.Mean <- 0
M.Up.Sigma <- 0.7
M.Down.Mean <- 0
M.Down.Sigma <- 0.7
A.Mean <- 7
A.Sigma <- 3
pUp <- 0.40
pDown <- 0.20
ratioThreshold <- 1
minLib <- 15*10^6
maxLib <- 25*10^6
size <- 10

#-----
# Simulation of expression values (mu1 and mu2)
#-----


DEGenes <- sample(-1:1,G,TRUE,c(pDown,1-pDown-pUp,pUp))

nDown <- sum(DEGenes== -1)
nStables <- sum(DEGenes== 0)
nUp <- sum(DEGenes== 1)

M <- rep(0,G)
M[DEGenes==1] <- ratioThreshold+exp(rnorm(nUp,M.Up.Mean,M.Up.Sigma))
M[DEGenes== -1] <- -ratioThreshold-exp(rnorm(nDown,M.Down.Mean,M.Down.Sigma))

A <- rnorm(G,A.Mean,A.Sigma)

muCond1 <- round(2^( (2*A-M)/2 ))
muCond2 <- round(2^( (2*A+M)/2 ))

M <- log2(muCond2)-log2(muCond1)
A <- (log2(muCond1)+log2(muCond2))/2

DEGenes <- sign(M)

```

```

DEGenes[is.na(DEGenes)] <- 0

round(summary(as.factor(DEGenes))/G*100)

# MA-plot with simulated expression values

jpeg("Fig01.jpg",640)
par(mar=c(4.1,4.1,0.1,0.1))
plot(A,M,pch=16)
points(A[M<0],M[M<0],pch=16,col="red")
points(A[M>0],M[M>0],pch=16,col="green")
points(A[M==0],M[M==0],pch=16,col="orange")
dev.off()

#-----
# Simulation of raw Counts
#-----

transcriptome1Size <- sum(muCond1)
transcriptome1Size
transcriptome2Size <- sum(muCond2)
transcriptome2Size

proba1 <- muCond1/transcriptome1Size
proba2 <- muCond2/transcriptome2Size

librarySize <- runif(6,minLib,maxLib)
librarySize

exp11 <- proba1*librarySize[1]
exp12 <- proba1*librarySize[2]
exp13 <- proba1*librarySize[3]
exp21 <- proba2*librarySize[4]
exp22 <- proba2*librarySize[5]
exp23 <- proba2*librarySize[6]

rawCounts <- matrix(nrow=G,ncol=6)
dimnames(rawCounts) <- list(1:G,c("A1","A2","A3","B1","B2","B3"))

for (i in 1:G) {
  rawCounts[i,1] <- rnbinom(1,size,,exp11[i])
  rawCounts[i,2] <- rnbinom(1,size,,exp12[i])
  rawCounts[i,3] <- rnbinom(1,size,,exp13[i])
  rawCounts[i,4] <- rnbinom(1,size,,exp21[i])
  rawCounts[i,5] <- rnbinom(1,size,,exp22[i])
  rawCounts[i,6] <- rnbinom(1,size,,exp23[i])
}

apply(rawCounts,2,sum)

# MA-plot with raw count means

meanCond1 <- apply(rawCounts[,1:3],1,mean)
meanCond2 <- apply(rawCounts[,4:6],1,mean)

```

```

M2 <- log2(meanCond2)-log2(meanCond1)
A2 <- (log2(meanCond1)+log2(meanCond2))/2

jpeg("Fig02.jpg",640)
par(mfrow=c(1,3),mar=c(4.1,4.1,0.1,0.1))
plot(A2,M2,pch=16,xlab="Mean of A values",ylab="Mean of M values")
points(A2[M<0],M2[M<0],pch=16,col="red")
plot(A2,M2,pch=16,xlab="Mean of A values",ylab="Mean of M values")
points(A2[M>0],M2[M>0],pch=16,col="green")
plot(A2,M2,pch=16,xlab="Mean of A values",ylab="Mean of M values")
points(A2[M==0],M2[M==0],pch=16,col="orange")
dev.off()

#-----
# Export of simulated data
#-----

simRawCounts <- cbind(muCond1,muCond2,M,DEGenes,rawCounts)

write.table(simRawCounts,"simRawCounts.txt",sep="\t",row.names=FALSE)

#-----
# The End
#-----

```

```

#-----
#
# Comparison of Normalization Methods for Differential Gene Expression Analysis
# in RNA-Seq Experiments: a Matter of Relative Size of Studied Transcriptomes.
#
#           - Supplementary Materials -
#
#           E. MAZA, P. FRASSE, P. SENIN, M. BOUZAYEN, M. ZOUINE
#
#           Supplementary File 2 (out of 3).
#
#-----

#-----
# Packages
#-----

library(DESeq)
library(edgeR)

#-----
# Data import
#-----

simRawCounts <- as.matrix(read.table("simRawCounts.txt",header=TRUE))

rawCounts <- simRawCounts[,5:10]
dimnames(rawCounts)[[2]] <- c("A1","A2","A3","B1","B2","B3")

nGenes <- dim(rawCounts)[1]

geneLength <- rep(1000,nGenes)

dim(rawCounts)

summary(rawCounts)

#-----
# Normalizations + DE test
#-----

cds <- newCountDataSet(rawCounts,rep(c("A","B"),each=3))

NumbOfSamples <- 6

# No Normalization

cds.noNorm <- cds
sizeFactors(cds.noNorm) <- rep(1,NumbOfSamples)
cds.noNorm <- estimateDispersions(cds.noNorm,"per-condition","maximum","local")
test.noNorm <- nbinomTest(cds.noNorm,"A","B")
write.table(test.noNorm,"test.noNorm.txt",sep="\t",dec=",",row.names=FALSE)

# Total counts normalization

```

```

cds.toCounts <- cds
totalCounts <- colSums(rawCounts)
normFactors <- totalCounts/exp(mean(log(totalCounts)))
sizeFactors(cds.toCounts) <- normFactors
cds.toCounts <- estimateDispersions(cds.toCounts,"per-condition","maximum","local")
test.toCounts <- nbinomTest(cds.toCounts,"A","B")
write.table(test.toCounts,"test.toCounts.txt",sep="\t",dec=",",row.names=FALSE)

# FPKM normalization

FPKM <- rawCounts
for (i in 1:NumbOfSamples)
    FPKM[,i] <- rawCounts[,i]/geneLength/totalCount[i]*10^9

FPKM <- ceiling(FPKM)

cds.FPKM <- newCountDataSet(FPKM,rep(c("A","B"),each=3))
sizeFactors(cds.FPKM) <- rep(1,NumbOfSamples)
cds.FPKM <- estimateDispersions(cds.FPKM,"per-condition","maximum","local")
test.FPKM <- nbinomTest(cds.FPKM,"A","B")
write.table(test.FPKM,"test.FPKM.txt",sep="\t",dec=",",row.names=FALSE)

# Upper Quartile normalization

cds.upQuartile <- cds
sizeFactors(cds.upQuartile) <- calcNormFactors(rawCounts,"upperquartile",p=0.75)
cds.upQuartile <- estimateDispersions(cds.upQuartile,"per-condition","maximum","local")
test.upQuartile <- nbinomTest(cds.upQuartile,"A","B")
write.table(test.upQuartile,"test.upQuartile.txt",sep="\t",dec=",",row.names=FALSE)

# Median normalization

cds.median <- cds
sizeFactors(cds.median) <- calcNormFactors(rawCounts,"upperquartile",p=0.5)
cds.median <- estimateDispersions(cds.median,"per-condition","maximum","local")
test.median <- nbinomTest(cds.median,"A","B")
write.table(test.median,"test.median.txt",sep="\t",dec=",",row.names=FALSE)

# TMM normalization

cds.TMM <- cds
sizeFactors(cds.TMM) <- calcNormFactors(rawCounts,"TMM")
cds.TMM <- estimateDispersions(cds.TMM,"per-condition","maximum","local")
test.TMM <- nbinomTest(cds.TMM,"A","B")
write.table(test.TMM,"test.TMM.txt",sep="\t",dec=",",row.names=FALSE)

# TMM50 normalization

cds.TMM50 <- cds
sizeFactors(cds.TMM50) <- calcNormFactors(rawCounts,"TMM",logratioTrim=0.49)
cds.TMM50 <- estimateDispersions(cds.TMM50,"per-condition","maximum","local")
test.TMM50 <- nbinomTest(cds.TMM50,"A","B")
write.table(test.TMM50,"test.TMM50.txt",sep="\t",dec=",",row.names=FALSE)

```

```

# RLE normalization

cds.RLE <- cds
sizeFactors(cds.RLE) <- calcNormFactors(rawCounts, "RLE")
cds.RLE <- estimateDispersions(cds.RLE, "per-condition", "maximum", "local")
test.RLE <- nbinomTest(cds.RLE, "A", "B")
write.table(test.RLE, "test.RLE.txt", sep="\t", dec=",", row.names=FALSE)

# Median ratio normalization

meanA <- apply(rawCounts[,1:3] %*% diag(1/totalCounts[1:3]), 1, mean)
meanB <- apply(rawCounts[,4:6] %*% diag(1/totalCounts[4:6]), 1, mean)

meanA1 <- meanA[meanA>0 & meanB>0]
meanB1 <- meanB[meanA>0 & meanB>0]

Ratios <- meanB1/meanA1

ratioMedian <- median(Ratios)
ratioMedian

normFactors <- c(rep(1,3), rep(ratioMedian,3))*totalCounts
normFactors <- normFactors/exp(mean(log(normFactors)))
normFactors

cds.medianRatio <- cds
sizeFactors(cds.medianRatio) <- normFactors
cds.medianRatio <- estimateDispersions(cds.medianRatio, "per-condition", "maximum", "local")
test.medianRatio <- nbinomTest(cds.medianRatio, "A", "B")
write.table(test.medianRatio, "test.medianRatio.txt", sep="\t", dec=",", row.names=FALSE)

#-----
# The End
#-----

```

```

#-----
#
# Comparison of Normalization Methods for Differential Gene Expression Analysis
# in RNA-Seq Experiments: a Matter of Relative Size of Studied Transcriptomes.
#
#           - Supplementary Materials -
#
#           E. MAZA, P. FRASSE, P. SENIN, M. BOUZAYEN, M. ZOUINE
#
#           Supplementary File 3 (out of 3).
#
#-----

#-----
# Results import
#-----

simRawCounts <- read.table("simRawCounts.txt",sep="\t",header=TRUE)

test.noNorm <- read.table("test.noNorm.txt",header=TRUE,sep="\t",dec=",")
test.toCounts <- read.table("test.toCounts.txt",header=TRUE,sep="\t",dec=",")
test.FPKM <- read.table("test.FPKM.txt",header=TRUE,sep="\t",dec=",")
test.upQuartile <- read.table("test.upQuartile.txt",header=TRUE,sep="\t",dec=",")
test.median <- read.table("test.median.txt",header=TRUE,sep="\t",dec=",")
test.TMM <- read.table("test.TMM.txt",header=TRUE,sep="\t",dec=",")
test.TMM50 <- read.table("test.TMM50.txt",header=TRUE,sep="\t",dec=",")
test.RLE <- read.table("test.RLE.txt",header=TRUE,sep="\t",dec=",")
test.medianRatio <- read.table("test.medianRatio.txt",header=TRUE,sep="\t",dec=",")

#-----
# DE Genes
#-----


alpha <- 0.05

DEGenes <- cbind(as.numeric(test.noNorm$padj<alpha)*sign(test.noNorm$log2FoldChange),
                  as.numeric(test.toCounts$padj<alpha)*sign(test.toCounts$log2FoldChange),
                  as.numeric(test.FPKM$padj<alpha)*sign(test.FPKM$log2FoldChange),
                  as.numeric(test.upQuartile$padj<alpha)*sign(test.upQuartile
$log2FoldChange),
                  as.numeric(test.median$padj<alpha)*sign(test.median$log2FoldChange),
                  as.numeric(test.TMM$padj<alpha)*sign(test.TMM$log2FoldChange),
                  as.numeric(test.TMM50$padj<alpha)*sign(test.TMM50$log2FoldChange),
                  as.numeric(test.RLE$padj<alpha)*sign(test.RLE$log2FoldChange),
                  as.numeric(test.medianRatio$padj<alpha)*sign(test.medianRatio
$log2FoldChange))

dimnames(DEGenes)[[2]] <- c("NoNo", "ToCo", "FPKM", "UpQu", "Medi", "TMM", "TMM50", "RLE", "MRN")

#-----
# hclust with simulated DE genes
#-----


withSimDEGenes <- cbind(DEGenes,simRawCounts$DEGenes)

```

```

dimnames(withSimDEGenes)[[2]] <-
c("NoNo", "ToCo", "FPKM", "UpQu", "Medi", "TMM", "TMM50", "RLE", "MRN", "Sims")
withSimEucliDist <- dist(t(withSimDEGenes))
withSimNormaClust <- hclust(withSimEucliDist)

jpeg("Fig03.jpg", 640)
p1clust(withSimNormaClust, xlab="Euclidean distance")
dev.off()

#-----
# False discoveries
#-----

TabRecap <- list(length=9)
for (i in 1:9)
  TabRecap[[i]] <- table(DEGenes[,i],simRawCounts$DEGenes)
nFaux <- matrix(nrow=3,ncol=9)
dimnames(nFaux)[[1]] <- c("False down-regulated","False non DE","False up-regulated")
dimnames(nFaux)[[2]] <- dimnames(DEGenes)[[2]]
for (i in 1:9) {
  nFaux[1,i] <- TabRecap[[i]][1,2]+TabRecap[[i]][1,3]
  nFaux[2,i] <- TabRecap[[i]][2,1]+TabRecap[[i]][2,3]
  nFaux[3,i] <- TabRecap[[i]][3,1]+TabRecap[[i]][3,2]
}

jpeg("Fig04.jpg", 640)
par(mar=c(2.1,2.1,0.1,0.1))
barplot(nFaux,col=c("red","orange","green"),legend=TRUE)
dev.off()

#-----
# The End
#-----

```